
실시간 트래픽 전송을 위한 RTP/RTCP의 TFRC 흐름제어 기법

TFRC Flow Control Mechanism based on RTP/RTCP for Real-time Traffic Transmission

최현아, 송복섭, 김정호

한밭대학교 정보통신전문대학원 컴퓨터공학과

Hyun-Ah Choi(hachoi@hanbat.ac.kr), Bok-Sob Song(serve73@hanbat.ac.kr),
Jeong-Ho Kim(jhkim@hanbat.ac.kr)

요약

지연시간으로 인한 부정확한 네트워크 상태정보로 발생하는 문제점을 해결하기 위해 단방향 지연시간을 이용하여 정확한 지연정보로 네트워크 상황을 판단하고 TFRC 흐름제어를 기반으로 네트워크 상태변화에 따라 실시간 멀티미디어 데이터의 전송율을 적절하게 조절할 수 있는 흐름제어 기법을 제안한다.

본 논문의 시뮬레이션 결과, 네트워크 상태 정보 측정을 위해 RTT와 OWD×2를 비교하여 평균적으로 약 12% 차이를 보였다. RTT를 사용하였을 경우 TFRC가 TCP보다 약 32%정도 더 많은 대역폭을 사용하였으며, OWD를 사용하였을 경우 3%정도의 오차를 보이고 링크의 공정한 대역폭을 사용할 수 있다. 네트워크 상황을 판단하고 TFRC 기반으로 네트워크 상태변화에 따라 TCP와 공정한 대역폭을 사용하면 실시간 데이터 전송에 맞게 전송율을 적절하게 조절하여 사용자들에게 고품질의 서비스를 제공할 수 있다.

■ 중심어 : | TFRC 흐름제어|RTP|RTCP | 실시간트래픽 |

Abstract

In this paper, to resolve the problem caused by a network state information inaccuracy the slow delay time that conclusion of network state of one way delay time which accuracy delay time information, according to network state changes on the based TFRC flow control, and suggest that flow control mechanism to adjust transfer rate fit of real time multimedia data. In simulation, to measure of network state information that on the average about 12% difference of compared RTT and OWD×2. When used RTT, used fair bandwidth TFRC much better than TCP about 32%, when used OWD, difference about 3% used fair bandwidth. Thus, conclusion of accuracy network state that used fair bandwidth according to network state changes on the based TFRC, users can support service of high quality that flow control mechanism to adjust transfer rate fit of real time data.

■ keyword : | TFRC Flow Control|RTP|RTCP|Real Time Traffic |

I. 서론

실시간 전송 트래픽은 대부분 VoIP, 영상의 등과 같은 지터나 패킷유실, 전송지연에 매우 민감한 멀티미디어 데이터를 전송하므로 높은 대역폭과 낮은 전송지연을 필요로 한다. 실시간 멀티미디어 서비스를 신뢰성 있는 TCP로 전송하면 네트워크 상태의 급격한 변화와 손실되는 패킷에 대한 재전송으로 인해 지연이 크고 지연시간 간격의 변화로 네트워크 사용의 전체적인 효율을 저해할 수 있다. UDP의 경우 손실 패킷에 대해 어떤 처리도 해주지 않으므로 연속적인 재생으로 실시간성을 보장하는 대신 타이밍 정보를 가지고 있지 않고 실시간 데이터의 QoS(Quality of Service)를 보장할 수 있는 기능을 제공하지 않기 때문에 고품질의 데이터를 전송할 수 없다. 유동적인 네트워크 상태변화에 따른 패킷 손실과 패킷 지연시간으로 혼잡상태에 빠졌을 때 이에 대응하는 메커니즘이 없기 때문에 이러한 특성을 고려한 흐름제어 기법이 필요하다. 이에 대해 실시간 서비스를 제공해 주기 위해 RTP(Real-Time Transport Protocol)을 이용하여 실시간 데이터를 전송하고 RTCP(Real-Time Control Protocol)를 이용하여 송신자에게 주기적으로 네트워크 상황을 피드백하여 전송품질을 높일 수 있다[1]. RTP/RTCP에 의해 전송되는 지터, 전송지연, 패킷손실 등을 이용하여 네트워크 상태를 측정하고 이에 따라 전송 속도를 조절해 주는 TFRC(TCP Friendly Rate Control), RAP(Rate Adaptive Protocol), TLFC(TCP Like Flow Control)와 같은 TCP Friendly 흐름제어 기법이 사용되고 있다[2].

본 논문에서는 지연시간으로 인한 부정확한 네트워크 상태정보로 발생하는 문제를 해결하기 위해, 단방향 지연시간을 이용하였다. 정확한 지연정보로 네트워크 상태를 판단하고 TFRC 흐름제어를 기반으로 네트워크 상태변화에 따라 실시간 데이터의 전송률을 적절하게 조절할 수 있는 흐름제어 기법을 제안하였다. 정확한 네트워크 정보를 예측하여 TFRC 기반으로 효율적인 망의 흐름을 제어한 방식으로 망의 혼잡에 대처할 수 있는 적응력을 높일 수 있음을 확인하였다.

II. 관련연구

1. RTP(Real-Time Transport Protocol)

음성, 화상회의 등과 같은 실시간 멀티미디어 데이터 전송을 위해서 제정한 프로토콜이며 멀티캐스트 또는 유니캐스트 네트워크 환경에서 실시간 데이터를 전송하는데 적합한 단대단 전송기능을 제공한다. RTP는 자원 예약에 대한 내용은 다루지 않으며, 특히 데이터 전송(Timely Delivery), QoS보장, 순차전달과 같은 기능을 제공하지 않기 때문에 시퀀스 번호로 패킷손실여부를 확인하고 타임스탬프로 데이터의 재생시점을 결정하는 기능을 제공한다[1].

2. RTCP(Real-Time Control Protocol)

실시간 전송 제어 프로토콜인 RTCP는 세션에 참가한 모든 참가자들의 전송상태에 대한 정보를 주기적으로 전송하며 최소한의 제어기능과 대체 식별 기능을 제공하여 흐름제어가 가능하다. RTCP는 SR(Sender Report), RR(Receiver Report), SDES(Source Description), BYE(goodBYE) 그리고 APP(APPLICATION define)의 5가지 제어 메시지를 교환하여 다음과 같은 기능을 수행한다. 첫째, 데이터 전송상태에 대하여 수신자들에게 피드백하여 흐름제어를 할 수 있다. 둘째, CNAME(Canonical NAME)를 이용해서 SSRC(Source ID)가 다른 것과의 충돌로 인하여 바뀌더라도 같은 사용자임을 알 수 있게 해준다. 셋째, 각각의 사용자들이 주기적으로 RTCP 패킷을 받기 때문에 몇 명의 사용자가 있는지 알수 있고, 사용자의 수가 늘어남에 따라 RTCP 패킷 주기를 조절해야 한다. 넷째, 최소한의 세션정보와 사용자 정보를 전달한다[1].

네트워크 상태 정보를 주기적으로 전달하는 RTCP의 보고 메시지는 SR과 RR 메시지를 통해 패킷의 개수, 수신된 패킷의 개수, 송신자와 수신자의 평균 왕복시간 등의 정보를 송신자에게 보내게 된다. [그림 1]과 같은 형태로 RTCP 패킷을 전송한다. RTCP의 패킷 발생간격은 5초로 제한되어 있으며 세션내의 참가자가 점차 증가할수록 전송간격이 커지면 흐름제어를 하는 시간 간격 또한 커지기 때문에 타임스케일 문제가 발생하고

TCP 대부분의 대역폭을 RTP가 차지하게 된다.

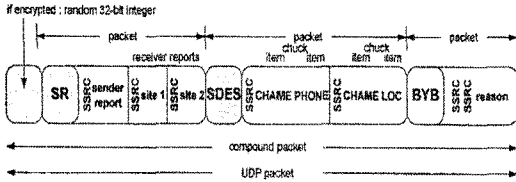


그림 1. RTCP 패킷 전송 형태

3. TCPF(TCP Friendly) 및 TFRC(TCP Friendly Control Rate) 흐름제어 기법

TCPF흐름제어 기법에서는 대표적으로 패킷 손실과 왕복지연시간(RTT:Round Trip Time) 두 가지 파라미터로 판단하여 네트워크 상태를 측정한다. 패킷손실을 통하여 네트워크 상태를 측정하면 수신자가 패킷이 손실되는 시점에서 지연시간을 추정하기 때문에 트래픽에 느린 대처로 부정확한 정보를 전달하게 된다. RTT는 순방향으로 데이터를 전송하고 역방향으로 RTCP 제어패킷이 네트워크 상태정보를 전달한다. 만약 역방향 지연시간 증가로 정확한 RTT값을 측정하기 어려워 사용 가능한 대역폭을 부정확하게 판단하게 된다. 결과적으로 충분한 대역폭이 남아있는데도 불필요하게 전송률을 떨어뜨림으로써 네트워크 사용의 전체적인 효율을 저해할 수 있고 또는 실제 상황과 다르게 많은 패킷을 전송하여 망의 혼잡상황을 유발하게 된다[3]. 부정확한 네트워크 상태정보로 발생하는 문제점을 해결하기 위해 RTT를 단방향 지연시간(OWD:One-Way Delay) 값을 이용해 정확한 상태정보를 측정할 수 있다. 하지만 단방향 지연시간 값을 구하기 위해서는 송수신자 사이의 시간 동기화 작업이 필요하다[4].

기존의 TCP 전송률을 모델링한 방정식을 기반으로 패킷 손실률과 패킷 지연시간을 이용하여 전송률을 조절하는 알고리즘이다. 식 1은 흐름제어를 위해 평균 전송율을 패킷 손실률과 전송지연과 같은 파라미터들을 사용하여 모델링하였다[2].

$$T = \frac{s}{R\sqrt{\frac{2p}{3}} + t_{RTO}(3\sqrt{\frac{3p}{8}})p(1+32p^2)} \quad (\text{식 1})$$

- R : 왕복시간 R
- RTO : 재전송 타임아웃
- s : 패킷크기 s
- p : 패킷 손실율 p

평균 전송률을 T를 구하여 네트워크 상황에 맞게 전송률을 조절하는데 패킷 손실율은 전송율의 선형적인 변화에 영향을 주며 RTT는 전송율 변화에 큰 영향을 미친다. TFRC는 패킷 전송지연 파라미터를 이용해서 수신측에 일정 주기로 네트워크 상태정보를 보내게 될 때 매 RTT마다 갱신하여 송신측에 전송하고 송신측은 피드백 된 정보를 통해 네트워크 상황에 맞는 전송율을 판단하여 패킷을 전송한다. TFRC에서 지원하는 파라미터는 대부분 패킷손실률에 초점을 맞추어 전송율을 산출하기 때문에 지연변화가 심한 환경에서는 정확한 지연정보를 측정할 수 없어서 전송률을 제어하는데 부정확하게 측정되어 네트워크 사용의 전체적인 효율을 저해할 수 있다[4].

UDP에 기반 하여 손실률을 최소화하면서 실시간 데이터를 전송하기 위해 제안되었다[5][6]. UDP에는 없는 혼잡제어 기능을 구현하기 위해 네트워크 상태를 측정하는 방정식을 이용한다. 방정식 기반의 혼잡제어의 목적은 이용 가능한 대역폭을 발견 이용하는 것이 아니라, 상대적으로 안정된 상태를 유지하면서 혼잡상태에 적응하는 것이다. 이러한 목적을 위해 현재 다수를 차지하는 Reno TCP의 모델링 방정식을 혼잡제어 알고리즘으로 이용한다[6].

$$T = \frac{s}{RTT\sqrt{\frac{2p}{3}} + RTO*\min(1,3\sqrt{\frac{3p}{8}})p(1+32p^2)} \quad (\text{식 2})$$

식 2의 TCP와 공정성을 유지하며 안정된 전송을 위해서는 다음과 같은 특성이 요구된다[7].

- ① 공격적으로 이용 가능한 대역폭을 찾지 않는다. 대신 손실 이벤트율(loss event rate)에 맞춰 전송률을 서서히 증가, 감소시킨다.
- ② 하나의 손실 이벤트(loss event)에 전송률을 반으로 줄이지 않는다. 그러나 몇 개의 연속적인 손실 이벤트에는 전송률을 반으로 줄인다.
- ③ 수신단은 임의 구간에서 패킷을 하나라도 받으면 RTT마다 적어도 한번의 피드백(feedback)을 전송하여야 한다.
- ④ 송신단은 몇 번의 RTT이후에도 피드백을 받지 않으면 전송률을 감소시키고 궁극적으로는 전송을 멈춘다.

III. 실시간 트래픽 전송을 위한 RTP/RTCP 흐름제어 기법

실시간 트래픽 전송을 위한 흐름제어는 RTCP SR(Sender Report)과 RR(Receiver Report) 메시지를 통해 네트워크 상태를 판단하여 전송률을 조절하게 된다. 네트워크 상태 정보를 측정하기 위한 파라미터로 송수신자의 왕복 지연시간을 측정할 값을 이용해서 파악하게 된다. RTT는 순방향으로 데이터 전송이 이루어지고 역방향으로 네트워크 상태 정보를 RTCP 제어 메시지로 보고된다. 만약 역방향이 패킷손실이나 혼잡 상태가 발생하게 되면 역방향의 지연시간은 증가하여 부정확한 네트워크 상황을 반영하여 망의 전체적인 효율을 저해하게 되는 문제점을 갖는다. 실시간 전송 시스템은 시간 지연에 매우 민감하므로 손실된 패킷에 대한 재전송으로 발생하는 지연시간으로 인해서 사용자들에게 고품질의 데이터를 제공하지 못한다. TFRC는 손실된 패킷에 대해 재전송 없이 다음 패킷을 전송하며 전송지연 시간을 줄여 고품질의 데이터를 제공할 수 있다. 지연시간으로 인한 부정확한 네트워크 상태정보로 발생하는 문제를 해결하기 위해 단방향 지연시간을 이용하여 정확한 지연정보로 네트워크 상황을 판단하고 TFRC 흐름제어를 기반으로 네트워크 상태변화에 따라 실시간 멀티미디어 데이터의 전송률을 적절하게 조

절할 수 있는 흐름제어 기법을 제안한다.

1. 단방향 지연시간을 이용한 네트워크 상태정보 측정

송신자와 수신자 사이의 단방향 지연 시간을 얻기 위해서는 시간 동기화(global time synchronization) 작업이 필요하나 시스템간에 서로 시간이 동기화되어 있지 않기 때문에 시간 동기화 작업을 위해 알고리즘을 사용하여 송신자와 수신자 사이의 시간을 동기화 시킨다[4]. [그림 2]는 단방향 지연시간을 구하기 위한 동기화 순서도를 도식화 한 것이며, 동기화 알고리즘을 이용하여 정확한 단방향 지연 시간을 측정할 수 있다[3].

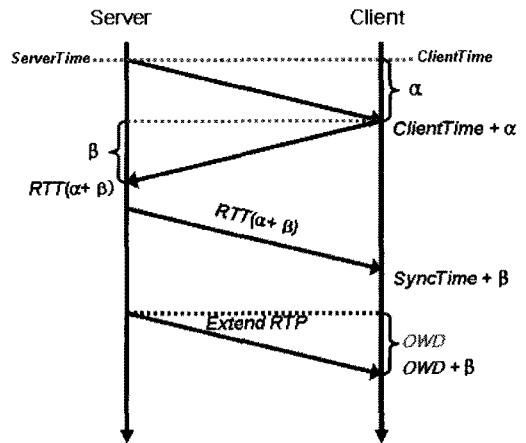


그림 2. 시간 동기화 순서도

송신측에서 수신측까지의 시간 α, 수신측에서 송신측까지의 시간 β라 하며 송신측에서 ServerTime을 수신측으로 보냈을때 패킷이 도착한 시점의 수신측 시간은 ClientTime+α이며, ServerTime을 저장한 후 송신측으로 되돌려준다.

$$(ClientTime + OWD + SyncTime) - ServerTime = OWD \quad (식 3)$$

단방향 전송지연 값인 OWD를 나타내며 식(3)을 통해 단방향 전송지연 값인 OWD를 구할 수 있다. 하지만

ClientTime을 알아낼 수 있는 방법이 없으므로 정확한 SynTime을 구할 수 없다.

$$\text{SyncTime} + \beta = \text{ServerTime} + \text{RTT} - (\text{ClientTime} + \alpha) \quad (\text{식 } 4)$$

$$(\text{ClientTime} + \text{OWD} + \text{SyncTime} + \beta) - \text{ServerTime} = \text{OWD} + \beta$$

정확한 SyncTime을 구하기 위해 식 4를 이용하여 OWD값을 구할 수 있으며 β 값 만큼 네트워크 상태 판단의 오차가 발생한다[3].

2. TFRC 기반 흐름제어의 적용

RTP/RTCP을 기반으로 수신측에서 주기적으로 전송하는 패킷 손실과 RTT 정보를 RTCP 제어 메시지로 보고한다. 수신측에서 보고된 정보를 바탕으로 네트워크 상태가 혼잡상태나 안정상태로 판단하며 각각의 상태에 따라 실시간 멀티미디어 데이터의 전송율을 증가시키거나 감소시키게 된다. 실시간 트래픽 전송을 위한 TFRC 기반의 데이터 전송율을 제어하는 기법은 [그림 3]과 같다. 송신측에서 RTP를 이용하여 실시간 멀티미디어 데이터들을 전송하면 수신측에서는 수신된 패킷들의 순서번호(sequence number)를 비교하여 패킷 손실 정보를 축적하고 타임스탬프(time stamp)로 RTT를 측정한다. 송신측에서 주기적으로 SR(Sender Report) 패킷을 전송하면 수신측에서는 그에 대한 응답으로 SR 패킷간에 발생한 패킷 손실의 누적 횟수(Nloss)와 SR 패킷 타임스탬프(LSR) 그리고 SR 패킷 처리 지연 시간(DLSR) 등의 정보를 담은 RR(Receiver Report) 패킷을 송신측에 전송한다.

송신측에서 RR 패킷을 수신하면 패킷 수신 시간(tRR)을 측정한다 다음, 식 5를 통해 패킷 손실률을 구한다. 여기서 패킷 손실률은 송신측에서 SR 패킷간에 전송한 총 패킷 수를 나타낸다.

$$p = \frac{N_{loss}}{N_{total}} \quad (\text{식 } 5)$$

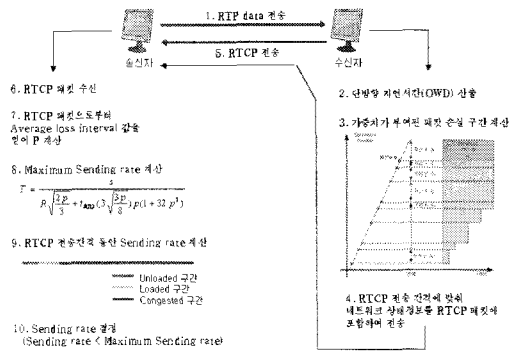


그림 3. TFRC의 동작원리

다음 식 6에서 RTT 대신 OWD 값을 적용하여 단방향 지연시간 정보를 구할 수 있다. 송신자가 측정하는 RTT와 수신자가 측정하는 RTT 그리고 순방향 지연시간의 초기값인 세 개의 값을 산출하게 된다. 는 송신자가 측정하는 RTT이며, 는 수신측에서 측정하는 RTT 이다 [4]. 다음 식 6을 적용한 시간 동기화 알고리즘을 통해 구현된다.

$$R = t_0 - \sum_{i=1}^n [RTT(s,i) - RTT(r,i)] \quad (\text{식 } 6)$$

송신측은 식 7과 같이 RTO(Retransmission Time-Out) 값을 계산한다.

$$t_{RTO} = R + 4 * RTT_{var} \quad (\text{식 } 7)$$

송신측은 식 5~식 7에서 산출된 각 파라미터를 이용하여 최종적으로 식 1을 이용하여 평균 전송율과 비교하여 전송율을 결정한다.

송신자는 최소 5초의 패킷 전송 간격으로 수신측에게 RTCP 제어 메시지를 통해 현재의 네트워크 상태를 보고 받는다. RTCP는 현재의 네트워크 상태정보를 측정하는데 필요한 파라미터들을 이용하여 식 1을 통해 최대 전송율을 구하게 되고 이 값을 넘지 않는 범위에서 전송율을 조절하게 된다. RTCP 패킷을 통하여 RTCP 패킷 전송간격 동안의 수신자가 측정하는 패킷손실률에 따라 임계값을 적용하여 다음 [표 1]과 같이 네트워크

상태를 Unload, Loaded, Congested 세 구간으로 나누어 전송률을 조절한다[8].

표 1. 네트워크 상태정보를 고려한 전송률 조절

구분	Sending rate
Unloaded	If Sending rate < Maximum rate $r_i = r_{i-1} + \frac{r_{i-1} - 1}{\pm \times RTT}$
	If Sending rate > Maximum rate $r_i = r_{i-1} - \frac{r_{i-1}}{\gamma}$
Loaded	$r_i = r_{i-1} - \frac{r_i}{\gamma}$
Congested	$r_i = \frac{r_{i-1}}{2}$

IV. 시뮬레이션 환경 및 성능분석

본 논문에서 사용한 Simulator는 버클리 대학의 NS-2(network simulator-2)를 사용하였다. RTT 대신 단방향 지연시간을 이용하여 TFRC 기반으로 전송율을 조절하는 흐름제어 기법에 대한 성능분석을 수행하였다.

[그림 4]는 성능평가를 위한 네트워크 구성이다.

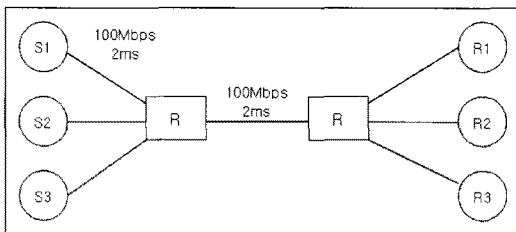


그림 4. 시뮬레이션 구성

1. 전송지연의 오차율 측정 성능분석

대칭 네트워크 토폴로지에서 수행한 시뮬레이션 결과로써 순방향 지연 시간은 송신자와 수신자의 클럭을 동기화시킨 후, 송신자의 타임 스탬프와 수신자의 클럭의 차이로 구했다. [그림 5]와 [표 2]에서 RTT와

OWD×2의 값을 비교하였다.

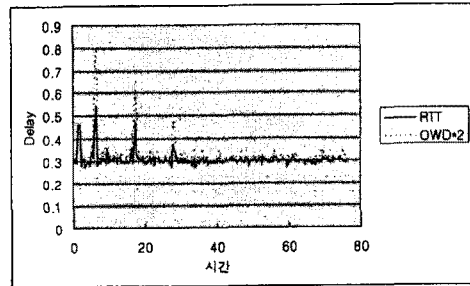


그림 5. RTT를 사용한 TFRC

표 2. 시뮬레이션을 하기위한 트래픽 특성

	RTT	OWD×2
평균	0.313906	0.346872

[그림 5]에서는 OWD의 값이 RTT에 비해 크게는 두 배의차이를 보이며 전반적으로 조금씩 차이를 보이는 모습을 볼 수 있다. 이는 지연 변화가 일정 수준 이상인 환경에서 RTT의 부정확한 정도를 나타낸다고 할 수 있다. [표 2]에서는 수집된 RTT와 OWD×2값들의 평균 값을 비교하면 평균적으로 10% 차이를 보이고 있으며 이 값을 기준으로 전송율을 조절할 때 부정확한 정보로 네트워크 효율을 떨어뜨릴 수 있다.

2. RTT와 OWD값을 측정된 전송율

다음 [표 3]은 성능평가를 위한 구성된 시뮬레이션 환경이다. 시뮬레이션을 수행하기 위해 경쟁하는 트래픽으로 TCP 트래픽과 TFRC 트래픽을 사용하였으며 제안하는 단방향 전송지연을 측정하기위한 RTP 트래픽을 사용하였다.

표 3. 시뮬레이션을 하기위한 트래픽 특성

Server-Client	Protocol	Application	Transmission Rate
S1 - R1	TCP	CBR	100Mbps
S2 - R2	TFRC	CBR	100Mbps

시뮬레이션 기준은 RTT를 이용하여 TFRC를 사용했을 때와 단방향 지연시간을 이용하여 TFRC를 사용

했을 때의 두 가지의 경우로 성능평가를 하였다. 각각 100초간 전송하여 RTT, 단방향 지연 시간, 전송율을 측정하고 결과를 비교하였다. [그림 6]은 네트워크 상태 정보를 파라미터를 RTT를 이용하여 TFRC와 TCP를 비교하였고, [그림 7]은 OWD를 이용하여 비교하였다.

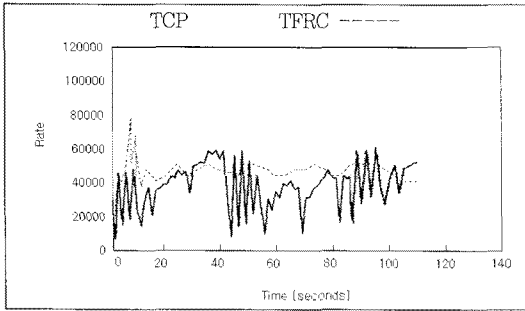


그림 6. RTT를 사용한 TFRC

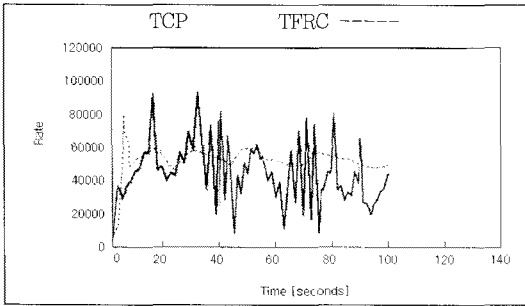


그림 7. OWD를 사용한 TFRC

[그림 6]과 [그림 7]에서 RTT를 사용하였을 때의 경우 TFRC가 TCP보다 약 32%정도 더 많은 대역폭을 점유하고 있었다. 이에 반해 OWD를 사용했을 때의 결과는 약 3%정도만이 오차를 보일 뿐 거의 같은 대역폭을 사용함으로써 OWD가 보다 정확하게 전송률을 예측하고 있음을 확인할 수 있었다. [표 4]에서 RTT와 OWD의 평균 전송율을 비교하였다.

표 4. RTT와 OWD의 평균 전송율 비교

delay time	TCP	TFRC	전송율 비교
RTT 사용	35230.96	48350.04	32%
OWD 사용	40175.05	40890.54	3%

V. 결론

본 논문에서는 지연시간으로 인한 부정확한 네트워크 상태정보로 발생하는 문제를 해결하기 위해 RTT 대신 단방향 지연시간을 사용하여 오차없는 지연정보를 얻고자 단방향 지연시간 알고리즘을 적용하여 정확하고 안정된 네트워크 상태정보를 측정할 수 있는 전송 흐름의 메카니즘을 제안한다.

제안한 단방향 지연시간을 이용하여 정확한 네트워크 정보를 예측하여 TFRC 기반으로 효율적인 망의 흐름을 제어한 방식으로 망의 혼잡에 대처할 수 있는 적응력을 높일 수 있음을 확인하였다. 이에 대한 시뮬레이션 실행함으로써 정확한 네트워크 상태정보 측정을 위해 RTT와 OWD의 경우로 기존의 인터넷망에서의 TCP와 TFRC의 공평한 대역폭의 사용을 비교하였다. RTT를 사용하였을 경우 TFRC가 TCP보다 약 32%정도 더 많은 대역폭을 사용하였으며, OWD를 사용하였을 경우 약 3%정도의 오차를 보이고 링크의 공정한 대역폭을 사용할 수 있다. 이러한 정보를 바탕으로 네트워크 상황을 판단하고 TFRC 기반으로 네트워크 상태변화에 따라 TCP와 공정한 대역폭을 사용하면 실시간 데이터 전송에 맞게 전송율을 적절하게 조절하여 실시간으로 사용자들에게 고품질의 서비스를 제공할 수 있다.

향후 연구과제로는 다양한 멀티미디어 서비스는 무선 네트워크 환경에서 많은 콘텐츠들이 제공될 것이며, 이에 무선망의 안정성과 변화량에 적절히 대응할 수 있는 유동적인 RTCP 기반의 흐름제어 기법이 필요하다.

참고 문헌

- [1] S. Casner, R. Frederick, and V. Jacobson, "RTP:A Transport Protocol for Real-Time Application," RFC1890, 1996(1).
- [2] M. Handley, S. Floyd, J. Padhye, and J. Widmer, "TCP Friendly Rate Control(TFRC):Protocol Specification," RFC3448, 2003(1).
- [3] 김용술, 김화성, "고품질 실시간 스트리밍 서비스

제공을 위한 네트워크 모니터링 기법”, 한국통신학회, 2006.

- [4] J. H. Choi and C. Yoo, "One-Way Delay Estimation and Its Application," Computer Communication, Vol.28, Issue 7, 2005.
- [5] S. Floyd, M. Handley, and J. Widmer, Jitendra Padhye, "Equation-Based Congestion Control for Unicast Application," Proceeding of ACM SIGCOMM'00, pp.43-56, 2000(8).
- [6] <http://www.aciri.org/tfrc/draft-ietf-tsvwg-tfrc-02.txt>
- [7] 이상철, 장주옥, "TFRC 프로토콜의 평균 손실 구간 계산방식의 비교평가, 정보과학회논문지, 제29권, 제5호, pp.495-500, 2002(5).
- [8] S. S. Ha, "TCP-friendly RTP rate control," 2003.
- [9] S. G. Na and J. S. Ahn, "TCP-like Flow Control Algorithm for Real-time Applications," IEEE ICON2000, 2000(9).
- [10] 나승구, 백갑천, 안중석, 김승범, "RTP 트래픽을 위한 효율적인 흐름제어 기법", 한국정보과학회 하계컴퓨터통신 워크숍, 1998.
- [11] K. Anagnostakis, M. Greenwald, and R. Ryger, "cing: Measuring network internal delays using only existing infrastructure," Proceedings of INFOCOM, 2003(4).
- [12] <http://www.isi.edu/nsnam/ns>.
- [13] D. Mills : Improved Algorithms for Synchronizing Computer Network Internal Delays using only Existing Infrastructure, In Proc. INFOCOM 2003, 2004(4).
- [14] S. Moon, P. Skelly, and D. Towsley : Estimation and Removal of Clock Skew from Network Delay Measurements, In Proc. INFOCOM 1999, 1999(3).

저자 소개

최 현 아(Hyun-Ah Choi)

준회원



- 2006년 2월 : 국립한밭대학교 컴퓨터공학과(공학사)
- 2008년 2월 : 국립한밭대학교 정보통신전문대학원 컴퓨터공학과(공학석사)

<관심분야> : 인터넷실시간서비스, 컴퓨터네트워크, 통신서비스

송 복 섭(Bok-Sob Song)

정회원



- 2005년 2월 : 국립한밭대학교 컴퓨터공학과(공학사)
- 2007년 2월 : 국립한밭대학교 정보통신전문대학원 컴퓨터공학과(공학석사)

• 2007년 3월 ~ 현재 : 국립한밭대학교 정보통신전문대학원 컴퓨터공학과 박사과정
<관심분야> : 데이터통신, 컴퓨터네트워크, 통신서비스, 정보보호

김 정 호(Jeong-Ho Kim)

종신회원



- 1980년 2월 : 경북대학교 공과대학 전자공학과(공학사)
- 1983년 2월 : 경북대학교 대학원 전자공학과(공학석사)
- 1994년 2월 : 단국대학교 대학원 컴퓨터공학과(공학박사)

- 1983년 ~ 1996년 : 한국전자통신연구소 책임연구원, 실장
- 1989년 : 정보처리기술사
- 1990년 : 공업계측제어기술사
- 1991년 : 정보통신기술사
- 1996년 ~ 현재 : 국립한밭대학교 정보통신·컴퓨터공학부 교수

<관심분야> : 데이터통신, 컴퓨터네트워크, 통신서비스